FACTA UNIVERSITATIS (NIŠ)
SER. MATH. INFORM. Vol. 30, No 1 (2015), 89–106

# MODIFIED GENETIC ALGORITHM WITH GREEDY HEURISTIC FOR CONTINUOUS AND DISCRETE P-MEDIAN PROBLEMS

**Lev A. Kazakovtsev, Victor I. Orlov,
Aljona A. Stupina and Vladimir L. Kazakovtsev**

**Abstract.** Genetic algorithm with greedy heuristic is an efficient method for solving large-scale location problems on networks. In addition, it can be adapted for solving continuous problems such as k-means. In this article, authors propose modifications to versions of this algorithm on both networks and continuous space improving its performance. The Probability Changing Method was used for initial seeding of the centers in case of the p-median problem on networks. Results are illustrated by numerical examples and practical experience of cluster analysis of semiconductor device production lots.

**Keywords**: Genetic algorithm; location problem; p-median problem; Probability Changing Method.

## 1. Introduction

The general aim of the optimal location problem is determining the location optimal of one or more new facilities in a continuous space or a discrete set when the number of possible locations is finite (discrete location problem) or infinite (continuous problem). The aim of a p-median problem on a network [16], which is one of the basic problems of discrete location theory, is finding $p$ nodes in a network such that the sum of distances from other nodes to the closest of the selected $p$ nodes is minimal. Search for medians is performed across the finite set of the nodes. In general, this problem is $\mathcal{NP}$-hard [27]. Polynomial time algorithm is developed for trees only [18]. In view of the high computational complexity of the problem, many heuristic algorithms were developed to solve this.

Despite the complexity of the problem, various heuristic algorithms give good results for most problems in reasonable time. One of the simplest but efficient methods for the $p$-median problem is local search [32, 31]. Rabbani [30] proposes an algorithm based on the new graph theory for small size problems. The p-median problems on bipartite graphs are considered [14], results are achieved

Received October 20, 2014.; Accepted January 13, 2015.
2010 *Mathematics Subject Classification.* Primary 90B85; Secondary 90C27, 90B80

for comparatively small-size problems, too. Using Lagrangian relaxation allows for an approximate solving of huge-scale problems [8, 7], up to 90000 vertices in a network. However, "good" solutions [7] were achieved by the analogous technique for problems with $n = 3795$ which were also considered as large-scale problems.

Heuristic methods do not guarantee finding an exact solution. However, they are statistically optimal. The percent of the problems which can be solved "almost optimal" grows with increase of the dimension of the problem [3, 4] (number of nodes $n$ and nodes to be selected $p$).

The idea of the genetic algorithms (GA) is based on a recombination of elements of some set of candidate solutions called "population". The candidate solutions are called "specimen". The first GA for solving the p-median problem was proposed by Hosage and Godschild [15]. Algorithm [11] gives rather precise results, however, its convergence is very slow. Alp, Erkut and Drezner [2] proposed a faster algorithm with a special "greedy" heuristic which is also precise. This algorithm for p-median problems was later improved by use of the probability changing method [5, 19] which was used for the initial seeding of $p$ vertices (centers) of a network.

The aim of the continuous p-median problem [39] is to find $p$ points (centers) such that the sum of distances from $n$ known points called demand points (or data vectors in case of the k-means problem) to the nearest of $p$ centers is minimal. The continuous location problems with Euclidean, Manhattan and Chebyshev metrics are well investigated, many algorithms based on Weiszfeld procedure and standard procedure for Manhattan metric are well known. In special cases, most complex continuous problems with restricted zones, barriers etc. can be converted into a discrete problem and solved approximately [20, 21, 33].

If the distances in a continuous problem are measured with use of the squared Euclidean metric, the p-median problem transforms into a k-means problem which is the most popular model of the cluster analysis [35, 26].

Most popular procedures for solving the continuous p-median problem (including k-means problem) are variations of the Alternating Location-Allocation procedure (ALA) [12]. This is a local search procedure which starts from some initial candidate solution. This initial solution is a set of $p$ points, called centers or centroids chosen among the demand points. Thus, choosing the initial solution is a discrete location problem.

The idea of using the genetic algorithm with the greedy heuristic [2] was proposed by Neema et al. [29]. In this case, the genetic algorithm is used for forming the initial solutions of the ALA algorithm which is performed at each step of the genetic algorithm.

In this paper, we propose new modification to such algorithms which increases its efficiency. Any ALA algorithm can be used with our new method. Results of running this modification for the continuous p-median problems was presented in [23]. Results are illustrated by numerical examples on standard testbeds and a practical example.

## 2. Known methods

Let $G = (V, E)$ be an undirected adjacent graph (a network), $V = \{v_1, ..., v_n\}$ be a set of its vertices, $E = \{e_i | i = \overline{1, m}\}$ be a set of its edges, $e_i = (v_j, v_k)$, $j \in \{\overline{1, n}\}$, $k \in \{\overline{1, n}\}$, $i \in \{\overline{1, m}\}$ without loops $(e_i \neq (v_j, v_j)\ \forall i = \overline{1, m}, j = \overline{1, n})$. For each edge $e_i$, its length $l_i$ is defined, $l_i \geq 0 \forall i = \overline{1, m}$. For an edge $e_i = (v_j, v_k)$, let us denote $l_{j,k} = l_i$. Weight $w_j \geq 0$ is defined for each vertex $v_j$. For each pair of the vertices $(v_j, v_k)$, a distance function $L(j, k)$ is defined as the length of the shordest path from $v_i$ to $v_j$.

$$L(j, k) = \min_{P \in P_{j,k}} \sum_{q \in P} l_q \tag{2.1}$$

Here, $P_{j,k}$ is a set of all possible paths between $v_j$ and $v_k$. We can formulate the the $p$-median problem as

$$\arg \min_{m_1, ..., m_p \in \{\overline{1,n}\}} f(m_1, ..., m_p) = \arg \min_{m_1, ... m_p \in \overline{1,n}} \sum_{i=1}^{n} w_i \min_{i=\overline{1,p}} L(m_j, i). \tag{2.2}$$

Let

$$C_i = \{k | \exists e_j = (c_i, v_k), j \in \{\overline{1, m}\}, k \in \{\overline{1, n}\}\}$$

be a set of the indexes of the vertices adjacent to the $i$th vertex.

For calculating the value of the objective function $f(m_1, ..., m_p)$, we can use the algorithm described in [5] or another algorithm.

For comparison, we used the local search (LS) [31] with random order of vertices evaluation (Algorithm 2.1) as one of the simplest and efficient algorithms [5].

**Algorithm 2.1.** Local search (LS)

**Require:** array of indexes $\mathcal{M} = \{m_1, ..., m_p\}$ of the vertices (initial solution), value of the objective function $f^* = f(m_1, ..., m_p)$.

**1:** shuffle elements of $\mathcal{M}$; $r = 0$;

**2:** for each element $m$ of the array $\mathcal{M}$ do

**2.1:** for each vertex $m^*$ which is adjacent to $m$ do

**2.1.1:** $f^{**} = f(m_1, ..., m^*, ..., m_p)$ (here, the vertex $m$ is replaced by $m^*$);

**2.1.2:** if $f^{**} < f^*$ then replace $m$ by $m^*$ in $\mathcal{M}$; $f^* = f^{**}$; $r = 1$;

**2.1.3:** next 2.1;

**2.2:** next 2;

**3:** if $r = 1$ then goto 1;

**4:** return new solution $(m_1, ..., m_p)$.

The GA with greedy heuristic proposed in [2] includes a special crossover procedure (Algorithm 2.2). The "chromosomes" of this algorithm are sets of the vertices (feasible solutions of the problem).

**Algorithm 2.2.** Crossover procedure for the GA with greedy heuristic on networks

**Require:** sets of vertex indexes $\mathcal{M}_1 = \{m_{1,1}, ..., m_{1,p}\}$, $\mathcal{M}_2 = \{m_{2,1}, ..., m_{2,p}\}$.

**1:** $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$;

**2:** while $|\mathcal{M}| > p$ do

**2.1:** $f* = +\infty$;

**2.2:** for each vertex $m^*$ in $\mathcal{M}$ do

**2.2.1:** $\mathcal{M}^* = \mathcal{M} \setminus \{m^*\}$;

**2.2.2:** $f** = f(\mathcal{M}*)$;

**2.2.2:** if $f^{**} < f^*$ then $m^{**} = m^*$;

**2.1.3:** next 2.2;

**2.3:** Set $\mathcal{M} = \mathcal{M} \setminus \{m^{**}\}$;

**2.3:** next 2;

**3:** return new solution ("chromosome") $\mathcal{M}$.

This method uses an original procedure of the initial population generation [2]. It does not use any mutation procedure.

In general, the continuous p-median problem can be formulated as

$$\arg \min_{X_1,...,X_p \in \mathbb{R}^d} \sum_{i=1}^{n} w_i \min_{j \in \{\overline{1,p}\}} L(X_j, A_i) \tag{2.3}$$

where $\{A_1, ..., A_n\}$ is a set of the demand points (data vectors) in a $d$-dimensional space, $w_1, ..., w_n$ are their weight coefficients (equal to 1 in case of the k-means problem), $X_1, ..., X_p$ are the points (centers) to be found and $L(\cdot)$ is some distance function (metric). In case of the squared Euclidean metric, $L(X_j, A_i) = \sum_{k=1}^{d} (x_{j,k} - a_{i,k})^2$, we have the k-means problem. Here, $X_j = (x_{j,1}, ..., x_{j,d}) \forall j = \overline{1,p}$, $A_i = (a_{i,1}, ..., a_{i,d}) \forall i = \overline{1,n}$.

One of the simplest ALA procedures known as the standard k-means procedure can be described as follows.

**Algorithm 2.3.** ALA procedure.

**Require:** initial centers $X_1, ..., X_p \in \{A_1, ..., A_n\}$.

**1:** For each demand point $A_i \in A_1, ..., A_n$, find the nearest center $C_i = \arg\min_{j=\overline{1,p}} L(A_i, X_j)$. Form $p$ sets (clusters) of the demand points closest to each of $p$ centers: $C_j^{clust} = \{i \in \{\overline{1,n} | C_i = j\}$;

**2:** For each cluster $C_j^{clust}$, $j = \overline{1,p}$, calculate its center $X_j$;

**3:** If Step 2 has changed at least one center then go to Step 1;

**4:** Otherwise, STOP.

In case of the squared Euclidean metric, searching for the new cluster center $X_j = (x_{j,1}, ..., x_{j,d})$ at Step 1 is a very simple problem:

$$x_{j,k} = \sum_{i \in C_j^{clust}} a_{i,k} / |C_j^{clust}| \forall k = \overline{1,d}.$$

In case of Euclidean metric, the new cluster center is a solution of the Weber problem [37], its approximate value can be found by the Weiszfeld procedure [38, 13]. To reduce the computational complexity, at Step 2, algorithm does not recalculate the centers of the clusters which have not been changed at Step 1.

Very efficient local search algorithms are further development of this standard procedure [1] in combination with other techniques such as sampling [35]. However, their result always depends on the initial solution. Known procedure k-means++ [6] improves the results in comparison with the chaotic chose of the initial solution and guarantees an approximation ratio $O(log(p))$ in expectation (over the randomness of the algorithm), where p is the number of clusters used[17].

The probability changing method initially proposed for unconstrained pseudo-Boolean optimization is a random search method. Its modifications for constrained problems proposed in [13] can solve problems with dimensions up to millions of Boolean variables.

The p-median problem can be solved using many methods [28, 35, 13] including the probability changing method [5]. We perform several steps of the probability changing method [5] and pass the results (its last population) to the GA with greedy heuristic as its initial population.

## 3. Original GA with greedy heuristic for continuous problems

To improve the accuracy of local search, many techniques of recombination of the initial sets of centers can be used. The GA with greedy heuristic was proposed for solving the p-median problems on networks [2]. Based on ideas of Algorithm 2.2 [2], Neema et al. [29] proposed its realization for the continuous p-median problem.

**Algorithm 3.1.** Genetic algorithm for the continuous p-median problem [29, 23].

**Require:** size of the population $N$, $n$ data vectors.

**1:** Form $N$ initial candidate solutions $\chi_1, ..., \chi_N \subset \{\overline{1,n}\}$. Here, each candidate solution is a set of indexes of data vectors, $|\chi_j| = p \ \ \forall j = \overline{1,N}$. Such initial solutions can be selected randomly or with use of the k-means++ procedure.

**2:** For each initial candidate solution, estimate the fitness function value $F_{fitness}(\chi_j)$ and store the values to variables $f_1, ..., f_N$. The ALA procedure with the initial solution $\{A_j | j \in \chi_i\}$ is performed to obtain the fitness function value $F_{fitness}(\chi_i) = \sum_{k=1}^{n} w_k \min_{j \in \{\overline{1,p}\}} L(X_j, A_k)$. Here, $X_1, ..., X_p$ are the centers obtained by the ALA procedure.

**3:** If the stop conditions are reached then STOP. The result of the algorithm is set $\chi_{i^*}$ such that $f_{i^*} = \min_{i \in \overline{1,N}} f_i$. For finding the final solution, the ALA procedure is performed again.

**4:** Select randomly two indexes $k_1, k_2 \in \{\overline{1,n}\}$, $k_1 \neq k_2$.

**5:** Form an interim solution $\chi_c = \chi_{k_1} \cup \chi_{k_2}$.

**6:** If $|\chi_c| \leq p$ then go to Step 9.

**7:** Calculate $j^* = \arg\min_{j \in \chi_c} F_{fitness}(\chi_c \setminus \{j\})$.

Here, to obtain the values of the fitness function, the ALA procedure is performed $|\chi_c|$ times.

**8:** Exclude $j^*$ from $\chi_c$: $\chi_c = \chi_c \setminus \{j^*\}$. Go to Step 6.

**9:** If $\exists i \in \{\overline{1,N}\} : \chi_i = \chi_c$ then go to Step 3.

**10:** Choose an index $k_3 \in \{\overline{1,N}\}$. In paper [29], the method of choosing this index is not determined. The original method [2] for the p-median problem on networks chooses the index of the solution which has the worst (maximum) value of the fitness function. However, the authors [2] do not consider this method as the only possible way of selecting. We used the simplest tournament selection: the algorithm chooses randomly two indexes $k_4, k_5 \in \{\overline{1,N}\}$; if $f_{k_4} > f_{k_5}$ then $k_3 = k_4$; otherwise, $k_3 = k_5$. This method slightly improves the results in comparison with [2].

**11:** Replace $\chi_{k_3}$ and $f_{k_3}$: $\chi_{k_3} = \chi_c$; $f_{k_3} = F_{fitness}(\chi_c)$.

**12:** Go to Step 3.

## 4. Modified algorithms

Step 5 of Algorithm 3.1 produces an interim solution $\chi_c$. In general, this solution is not feasible: its cardinality is up to $2p$. At Steps 7–8, Algorithm excludes one member from $\chi_c$ until $|\chi_c| = p$. These steps demand many starts of the ALA procedure which is performed $|\chi_c|$ times in each iteration. Thus, the ALA procedure starts up to $2p + 2p - 1 + 2p - 2 + ... + p + 1$ times. In addition, the computational complexity of the ALA procedure depends on the number of centers $|\chi_c|$.

Various stop conditions (Step 3) can be used [2, 29, 5]. Algorithm can be stopped after a number of steps without result improvement or if the time limit is reached.

Other idea of global search in case of the p-median problem include improving the local search results by replacing some part of centers with randomly selected demand points [29]. We propose a combination of both ideas. In our modification of Algorithm 3.1, we add a random number $r$ of the demand points of some candidate solution to another candidate solution. The distribution of $r$ allows adding $1...p$ solutions and tends to adding small number of solutions. Then, $|\chi_c| - p$ demand points are eliminated.

Thus, Step 5 of Algorithm 3.1 is as follows.

**5.1:** Generate a random value $r_{init} \in [0; 1)$ with the uniform distribution (standard Random function).

**5.2:** Set $r = 1 + [(p-1) \cdot r_{init}^2]$.

**5.3:** From set $\chi_{k_2}$, select randomly a subset $\chi^*_{k_2}$ with cardinality $r$.

**5.3:** From an interim solution $\chi_c = \chi_{k_1} \cup \chi^*_{k_2}$, select randomly a subset.

Here, the cardinality $r \in \{\overline{1, p}\}$ of set $\chi^*_{k_2}$ can be chosen randomly with the uniform distribution. However, the equation in Step 5.2 gives better results.

$\chi^b$ and $\chi^w$ are the best and the worst samples of the sets of vertex indexes $\chi$ generated by

Several modifications to Algorithm 3.1 were proposed in [22]. The modifications proposed here can be used with modifications proposed in [22] or separately.

For problems on networks, we propose the following modification of Step 1 of Algorithm 2.2.

**1.1:** Generate a random value $r_{init} \in [0; 1)$ with uniform distribution;

**1.2:** Set $r = 1 + [(p-1) \cdot r_{init}]$;

**1.3:** From set $\mathcal{M}_2$, select randomly a subset $\mathcal{M}^*_2$ with cardinality $r$;

**1.4:** Form an interim joint solution $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}^*_2$.

Results of proposed modifications of GAs with greedy heuristics on both networks and continuous spaces are explained below.

## 5. Numerical experiments

For testing purposes, we used data sets from the UCI library [40] and automatically generated data sets which are sets of pairs of uniformly distributed coordinates in a square 10x10.

The maximum number of data vectors in the k-means problems was 169309 ($n = 169309$, "Europe" data set from the UCI library), the maximum dimension was $d = 32$. In addition, we solved several p-median problems with Euclidean metric.

An important parameter of Algorithm 3.1 is the size of the population $N$. Neema et al. [29] do not propose any method of determining this value. In the original method for the p-median problems on networks [2], the size of the genetic algorithm population depends on the number of network nodes $n$ and $p$. However, unlike the p-median problem on a network, in case of the the continuous problem, computational experiments did not reveal any correlation of the problem size and the population size $N$ providing the fastest convergence. The optimal values of $N$ are larger for the large-scale problems than the optimal values for the small problems. However, the values of $N$ providing the fastest convergence and the most accurate results belong to the set $\{\overline{10, 27}\}$ for all tested problems ($n \in \{\overline{150, 169309}\}$, $p \in \{\overline{3, 100}\}$) for both, original Algorithm 3.1 and our modification. The smaller sizes

of the population reduce the accuracy, the larger values increase the time needed for obtaining the most accurate results. In all experiments below, we used populations of $N = 20$ candidate solutions. The results of computational experiments with various population sizes on the 2-dimensional generated data set, $n = 10000$, $p = 100$ are shown in Fig. 5.1. The average results for 20 runs are shown.

Computational experiments were performed on a system with CPU Xeon 5650 2.76 GHz, 12 Gb RAM, HyperThreading disabled. The GNU Fortran compiler was used.

The computational experiments were organized as follows. The original Algorithm 3.1 ran 20 times with time limit as the stop condition. We fixed the average fitness function value reached by the algorithm. Then, the original and modified algorithms ran more 20 times until reaching the fixed fitness function value as the new stop condition. In addition, we ran the algorithm which performed multiple starts of the k-means++ procedure.

For all problems with $n > 1000$ and $p > 5$, our modification of Algorithm 3.1 was faster than the original version of this algorithm [2, 29] with equal or higher accuracy. In Fig.5.4 and 5.2, we show the average results achieved by our algorithm and original algorithm fixed after each iteration.

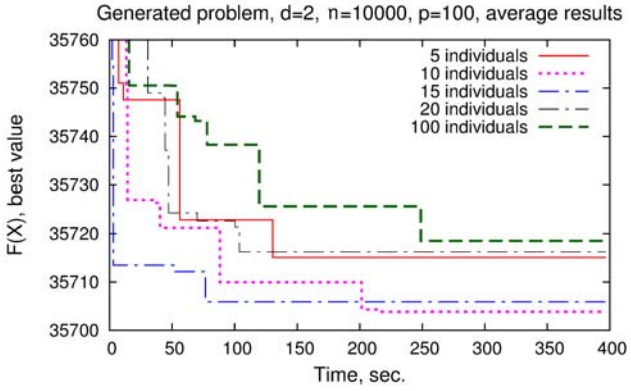



FIG. 5.1: Comparison of the results for modified GA with greedy heuristic for the planar p-median problem using various population sizes

The results of the modified GA with greedy heuristic on networks are shown in Table 5.1 and Fig. 5.3. The speed-up of our modification for networks is not as valuable as the speed-up of our modification for continuous problem.

For testing purposes, we used the local search method (Algorithm 2.1) with multistart from randomly generated initial solution as one of the simplest methods and the genetic algorithm (GA) [2] with the crossover procedure given by Algorithm 2.2 (greedy heuristic). As a testbed, we used the p-median problems from the OR Library [9]. This library contains problems with numbers of vertices up to $n = 900$. We used special algorithm [5] for generating larger problems.

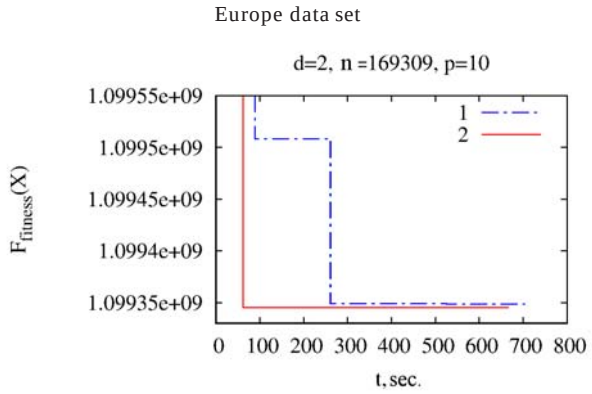



FIG. 5.2: Comparison of the results of the original and modified genetic algorithm with greedy heuristic for the planar p-median problem. 1 – original genetic algorithm with greedy heuristic, 2 – modified algorithm

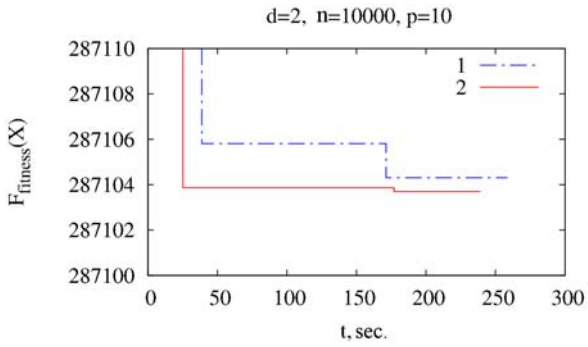
Fig. 5.3 shows the average values for 10 runs and the worst result. To calculate the quantity of exemplars of the generated solutions in each population $N$, we used formula

$$N = \lceil \frac{\sqrt{n}C\left( \begin{array}{c} n \\ p \end{array} \right)}{100\lceil n/p \rceil} \rceil \lceil n/p \rceil. \tag{5.1}$$

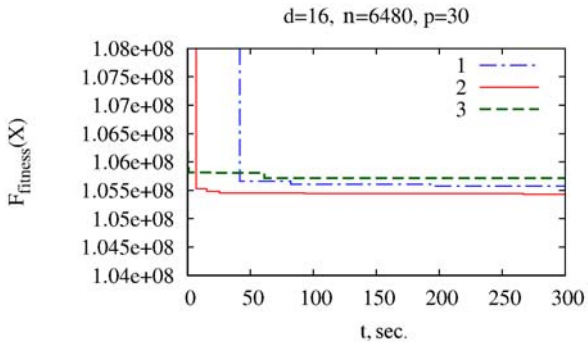
Table 5.1: Comparison of results for problems on networks

| Problem | Method | avg.time, sec. | Avg.result |
|---|---|---|---|
| Generated (n=2000, p=100) | LS multistart | 27.5 | 129440.66 |
| | Original GA | 73.76 | 120031.2 |
| | Original GA + LS | 4.4 | 119885.61 |
| | Modified GA | 53.52 | 120258.03 |
| | Modified GA + LS | 3.4 | 119865.35 |
| pmed11 (n=300, p=5) | LS multistart | 0.1 | 7578 |
| | Original GA | 0.7 | 7578,89 |
| | Original GA + local search | 0.05 | 7578 |
| | Modified GA | 0.42 | 7602.11 |
| | Modified GA + LS | 0.19 | 7578 |
| pmed13 (n=300, p=30) | LS multistart | 1.15 | 4311 |
| | Original GA | 13.25 | 4331 |
| | Original GA + LS | 0,28 | 4314.25 |
| | Modified GA | 0.52 | 4386.13 |
| | Modified GA + LS | 0.24 | 4311 |
| pmed17 (n=400, p=10) | LS multistart | 2.41 | 6980 |
| | Original GA | 6.22 | 6982 |
| | Original GA + LS | 0.95 | 6980 |
| | Modified GA | 1 | 7044.6 |
| | Modified GA + LS | 1.08 | 6981.2 |
| pmed22 (n=500, p=10) | LS multistart | 7.09 | 8464 |
| | Original GA | 15.48 | 8464 |
| | Original GA + LS | 1.89 | 8464 |
| | Modified GA | 2.69 | 8500.71 |
| | Modified GA + LS | 0.98 | 8464 |

A) Randomly generated data set




B) MissAmerica1 data set




C) Iris data set

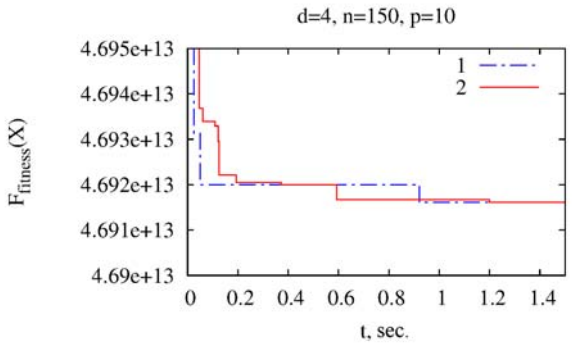



FIG. 5.3: Comparison of the results of the original and modified genetic algorithm with greedy heuristic for the k-means problems. 1 – original genetic algorithm with greedy heuristic, 2 – modified algorithm, 3 – k-means++ multistart

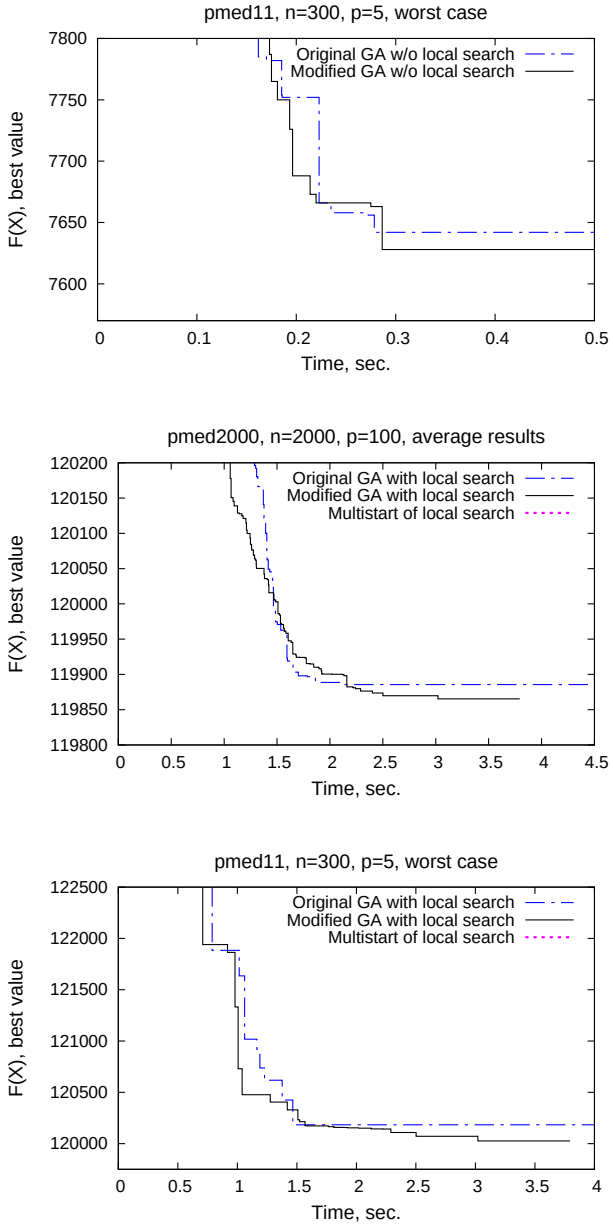



FIG. 5.4: Comparison of the results of the original and modified GA with greedy heuristic for the p-median problems on networks. Results of local search are outside the frame of the plots.

## 6. Practical example. Classification of the semiconductor devices

For any class of electronic production, using semiconductor devices with equal operational performance within one unit is preferred. The ideal situation is using semiconductor devices produced as a single production batch. At the same time, semiconductor device suppliers do not always guarantee the uniformity of the lot [24, 36]. In this case, complex testing of the supplied lots is the only way of improving quality and uniformity of the lots of the electronic elements. Cluster analysis of the tests results allows to estimate the uniformity.

Results of the ALA procedure depend on initial centers seeding. Thus, reproducibility of the results of an algorithm based on the ALA procedure is a serious problem. Information Bottleneck method of cluster analysis provides perfect reproducible results. However, such algorithms are very slow. GA with greedy heuristic are compromise methods providing precise results in reasonable time.

Various number of tests are performed depending on the class of the semiconductor device and the unit. In our example, 55 various values with various scale were measured for the electronic chip STK403-090. Proposed k-means algorithm was performed for classifying lots of this device on production batches produced under various conditions. Moreover, algorithm was performed for evaluation of the production bathes within one supplied lot of 700 devices.

MDS (Multi-Dimensional Scaling) [34, 10] and open-source visualization means ELKI [25] and GNUPLOT were used for visualizing of test data and their clusters.

We performed test data clustering for 10 lots of various devices (diods, stabilitrons, stabistors, transistors, chips) containing 1..7 production batches with under different production conditions. Each lot contains 60..1250 devices. Proposed algorithm was run for each lot 10 times with various value of $k \in \overline{\{1, 10\}}$ (estimated number of clusters representing production batches). Results of splitting of a lot of the chip STK403-090 into various number of clusters $k \in \overline{\{2, 3\}}$ are shown in Fig. 6.1 (MDS results). Result of running of our algorithm is the correspondence of the device identifiers in the lot and numbers of the clusters (estimated production batches). In addition, the result is the sum of distances from test data vectors and cluster centers in a normed space. Dependence of this total distance on cluster number $k$ (estimated production batches) is shown in Fig. 6.2.

In our tested lot, actual production batches are known. There are 3 actual production batches. Standard k-means procedure returns the correspondence with 6-10 errors (incorrect correspondence of cluster analysis results and actual production batches) after 10 algorithm runs (average value is 7.4 errors). Our algorithm reduces number of errors to 1-7 (average is 4.3).

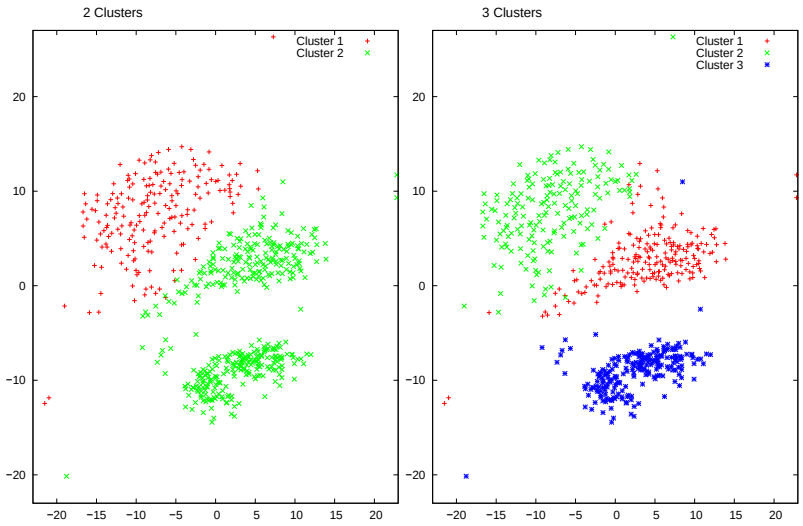



FIG. 6.1: Results of clustering of data of a lot of the chip STK403-090 into 2 and 3 clusters

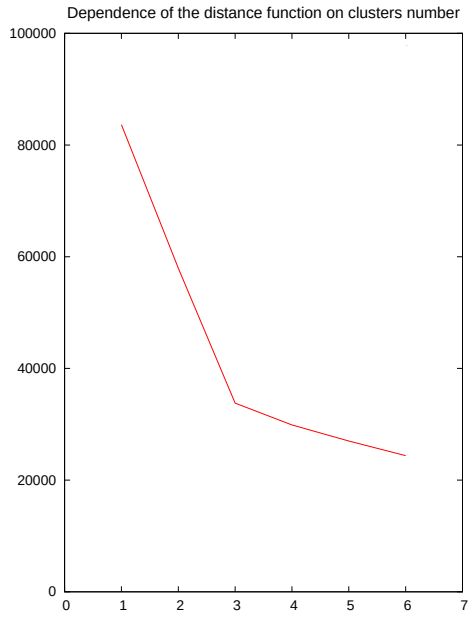



FIG. 6.2: Dependence of total distance on the number of clusters (estimated number of production batches of chip STK403-090). $d = 55$, $l_\infty$, $n = 700$.

## 7. Conclusion

The proposed modification to the genetic algorithm with greedy heuristic accelerates the convergence of the algorithm significantly. In our experiments, we were unable to find any continuous test problems which cannot be solved faster with our new algorithm when compare with the original algorithm with greedy heuristic.

The applicability of the proposed modification to the genetic algorithm with greedy heuristic for the p-median problems on graphs is subject to the further research.

The proposed modification to the GA with greedy heuristic for p-median problems on networks are less efficient. However, it improves the performance in most cases.

## REFERENCES


1. M. R. Ackermann et al: *StreamKM: A Clustering Algorithm for Data Streams*, J. Exp. Algorithmics, 2012, 17, Article 2.4 (May 2012), published online, DOI: 10.1145/2133803.2184450
2. O. Alp, E. Erkut and Z. Drezner: *An Ecient Genetic Algorithm for the p-Median Problem*, Annals of Operations Research, 2003, 122 (1-4), 21–42.
3. A. Antamoshkin: *On Optimal Algorithms of Optimization of Functionals with Boolean Variables*, In: Trans. Ninth Prague Conference on Inform. Theory, Statist. Dec. Functions, Random Processes. Academia, Prague 1983, 137–141.
4. A. Antamoshkin: *Optimization of functionals with Boolean variables*, Izd. Tomsk, un-ta, Tomsk 1987.
5. A. N. Antamoshkin and L. A. Kazakovtsev: *Random Search Algorithm for the p-median Problem*, Informatica, 2013, 37(3), 267–278.
6. D. Arthur and S. Vassilvitskii: *k-Means++: The Advantages of Careful Seeding*, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms, ser. SODA '07, 2007, 1027–1035.
7. P. Avella, A. Sassano and I. Vasilev: *Computational Study of Large-Scale p-Median Problems*, Mathematical Programming, 2007, 109(1), 89–114.
8. P. Avella, M. Boccia, S. Salerno and I. Vasilyev: *An Aggregation Heuristic for Large Scale p-median Problem*, Computers & Operations Research, 2012, 39 (7), 1625–1632, doi 10.1016/j.cor.2011.09.016
9. J. E. Beasley: *OR-Library: Distributing Test Problems by Electronic Mail, Journal of the Operational Research Society*, 1990, 41(11), 1069–1072.
10. J. F. P. Borg: *Modern Multidimensional Scaling: Theory and Applications*, Springer, 2005, 207–212.
11. B. Bozkaya, J. Zhang and E. Erkut: *A Genetic Algorithm for the p-Median Problem*, Drezner Z., Hamacher H. (eds.), Facility Location: Applications and Theory, Springer , 2002.
12. L. Cooper: *Location-allocation problem*, Oper. Res., 1963, 11, 331–343.

13. Z. Drezner: *The Fortified Weiszfeld Algorithm for Solving the Weber Problem*, IMA Journal of Management Mathematics, 2013, published online. DOI: 10.1093/imaman/dpt019

14. J. Fathali, N. J. Rad, S. R. Sherbaf: *The p-median and p-center Problems on Bipartite Graphs*, Iranian Journal of Matematical Sciences and Informatics, 2014, 9 (2), 37–43

15. C. M. Hosage and M. F. Goodchild: *Discrete Space Location-Allocation Solutions from Genetic Algorithms*, Annals of Operations Research, 1986, 6, 35–46.

16. S. L. Hakimi: *Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph*, Operations Research, May/June 1964, 12(3), 450–459.

17. T. Kanungo, D. Mount, N. Netanyahux, C. Piatko, R. Silverman, A. Wu: *A Local Search Approximation Algorithm for k-Means Clustering*, Computational Geometry, 2004, 28, 89112.

18. O. Kariv and S. L. Hakimi: *An Algorithmic Approach to Network Location Problems. II. The p-medians*, SIAM Journal on Applied Mathematics, 1979, 37(3), 539–560.

19. L. A. Kazakovtsev, A. N. Antamoshkin and M. N. Gudyma: *Parallelny algoritm dlya p-mediannoy zadachi (Parallel Algorithm for the p-Median Problem)*, Control Systems and Information Technologies, 2013, 52 (2.1), 124–128.

20. L. A. Kazakovtsev: *Adaptation of the Probability Changing Method for Weber Problem with an Arbitrary Metric*, Facta Universitatis (Niš) Ser. Math. Inform., V.27(2), 2012, 239–254.

21. L. Kazakovtsev: *Algorithm for Approximate Solution of the Generalized Weber Problem with an Arbitrary Metric*, Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on, Malta, 2012, 109–114.

22. L. A. Kazakovtsev and A. N. Antamoshkin: *Gnetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems*, Informatica, 2014, 38(3), 229–240

23. L. A. Kazakovtsev and A. A. Stupina: *Fast Genetic Algorithm with Greedy Heuristic for p-Median and k-Means Problems*, IEEE 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 6-8 October 2014, St.-Petersburg, 2014, 702–706

24. N. V. Koplyarova, V. I. Orlov, N. A. Sergeeva, V. V. Fedosov: *On Non-Parametric Models in the Problem of Performance Diagnostics of Electronic Components*, Zavodskaya Laboratoriya, 2014, 80(7), 37-77 (in Russian).

25. H. P. Kriegel, K. P. Kroeger and A. Zimek: *Outlier Detection Techniques (Tutorial)*, 13th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009), Bangkok, Thailand, 2009.

26. J. B. MacQueen: *Some Methods of Classification and Analysis of Multivariate Observations*, Proceedings of the 5th Berkley Symposium on Mathematical Statistics and Probability, 1967, Vol.1, 281–297.

27. S. Masuyama, T. Ibaraki and T. Hasegawa: *The Computational Complexity of the m-Center Problems on the Plane*, The Transactions of the Institute of Electronics and Comunication Engineers of Japan, 1981, 64E, 57–64

28. N. Mishra, D. Oblinger and L. Pitt: *Sublinear Time Approximate Clustering*, 12th SODA, 2001, 439–447.

29. M. N. Neema, K. M. Maniruzzaman and A. Ohgai: *New Genetic Algorithms Based Approaches to Continuous p-Median Problem*, Netw. Spat. Econ., 2011, 11,83–99, DOI:10.1007/s11067-008-9084-5.

30. M. Rabbani: *A Novel Approach for Solving a Constrained Location Allocation Problem*, International Journal of Industrial Engineering Computations, 2013, published online, doi 10.5267/j.ijiec.2013.02.003, http://www.growingscience.com/ijiec/IJIEC_2013_8.pdf

31. M. G. C. Resende, C. C. Ribeiro, F. Glover and R. Marti: *Scatter search and path-relinking: Fundamentals, advances, and applications*, Handbook of Metaheuristics, 2nd Edition, M. Gendreau and J.-Y. Potvin, Eds. Springer, 2010, 87–107

32. M. G. C. Resende: *Metaheuristic Hybridization with Greedy Randomized Adaptive Search Procedures*, in TutORials in Operations Research, Zhi-Long Chen and S. Raghavan (Eds.), INFORMS, 2008, 295–319

33. P. S. Stanimirovic, M. Ciric, L. A. Kazakovtsev and I. A. Osinuga: *Single-facility Weber Location Problem Based on the Lift Metric*, Facta Universitatis (Niš) Ser. Math. Inform., 2012, 27(2), 175–190.

34. Zh. Sun, G. Fox, W. Gu and Zh. Li: *A Parallel Clustering Method Combined Information Bottleneck Theory and Centroid Based Clustering*, The Journal of Supercomputing, 2014, 69(1), 452-467, DOI: 10.1007/s11227-014-1174-1.

35. P.-N. Tan, M. Steinbach and V. Kumar: *Cluster Analysis: Basic Concepts and Algorithms*, Chapter 8 / Introduction to Data Mining, Addison-Wesley, 2006, 487–567.

36. V. Subbotin and V. Steshenko: *Problemy obespecheniya bortovoy kosmicheskoy apparatury kosmicheskikh apparatov elektronnoy komponentnoy basoy (in Russian: Problems of Supply on-Board Space Equipment of Spaceships with Components)*, Komponenty i tekhnologii, 2011, 11, 10–12.

37. A. Weber: *Uber den Standort der Industrien, Erster Teil: Reine Theorie des Standortes*, 1922, Tubingen, Mohr.

38. E. Weiszfeld: *Sur le point sur lequel la somme des distances de n points donnes est minimum*, Tohoku Mathematical Journal, 1937, 43(1), 335–386.

39. G. Wesolowsky: *The Weber problem: History and perspectives*, Location Science, 1982, 1, 5–23.

40. *UCI Machine Learning Repository*, URL: https://archive.ics.uci.edu/ml/datasets.html

Lev A. Kazakovtsev
Siberian State Aerospace University Named after M. F. Reshetnev
prosp.Krasnoyarskiy Rabochiy, 31
660014 Krasnoyarsk, Russian Federation
Siberian Federal University
79 Svobodny pr., 660041 Krasnoyarsk, Russia
levk@bk.ru

Victor I. Orlov
TTC – NPO PM
ul. Molodezhnaya, 20
662970 Zheleznogorsk, Krasnoyarskii Krai, Russian Federation
ttc@krasmail.ru

Aljona A. Stupina

Siberian State Aerospace University Named after M. F. Reshetnev
prosp.Krasnoyarskiy Rabochiy, 31
660014 Krasnoyarsk, Russian Federation
Siberian Federal University
79 Svobodny pr., 660041 Krasnoyarsk, Russia
saa55@rambler.ru

Vladimir L. Kazakovtsev
Novosibirsk State University
Specialized Educational and Scientific Center (Distant School)
ul.Pirogova, 2
630090 Novosibirsk, Russian Federation
vokz@bk.ru